

Git bash

Repo local: Donde se encuentran los archivos de forma local en nuestro pc.
Commits: Son comentarios que guardaran que se ajusto, quien lo ajusto y en que momento lo hizo.

comandos:

```
ls -R Muestra las carpetas y sus contenidos
ls -l Muestra toda la información de una carpeta: usuario, grupo,
permisos, tamaño, fecha y hora de creación
ls -t Comando para ver la lista de archivos y los ultimos modificados o
trabajados
cp Comando para copiar archivos
rm Elimina un archivo
rmdir Elimina una carpeta vacia
rm -r Elimina una carpeta y su contenido
more Muestra el contenido de un archivo
ls -a Muestra todos los archivos ocultos)
mv Mueve archivos, renombra las carpetas (para renombrar ese nombre no
debe existir)
mv Mueve el contenido de carpeta1 a carpeta2
cp Copia un archivo o carpeta
" " Nos permiten utilizar términos que consistan en más de una palabra y
con espacios
. Permite hacer referencia al directorio donde estamos ubicados
actualmente
cd/name/name Cambia a una carpeta específica
git init Genera un repositorio local
git status Muestra los commits generados
git add "nombre del archivo" Para agregar los archivos modificados al
repositorio
git add . Agrega todos los archivos automaticamente
git commit -m Sirve para colocarle un comentario al commit que se va a
subir
git log Muestra todos los commits realizados
git remote add origin + URL de nuestro repositorio Sirve para conectar
git con github
git remote -v Sirve para verificar si nuestra conexion a nuestro
repositorio en github fue realizada con exito
git push Se utiliza para enviar archivos a nuestro repositorio en la nube
git push origin master Se utiliza para especificar a que rama se quieren
subir los ajustes
git clone + URLdelRepositorioRemoto Permite crear una copia exata de
todos los archivos que existen en el repositorio remoto. Solo se ejecuta
1 sola vez

git pull Baja de la nube o actualiza los archivos que se encuentran en el
repositorio o que hayan sufrido algun tipo de modificacion
git pull origin main Para indicarle que queremos traer todo el contenido
actualizado de la rama principal

git branch "nombreDeLaRama" Comando para crear una nueva rama en git
git branch Para ver las ramas creadas
git branch -d "nombreDeLaRama" Elimina la rama que se especifica. Git
evita que eliminemos la rama si tiene cambios que aún no se han fusionado
con la rama Main.
```

git branch -D Fuerza la eliminacion de la rama especificada, incluso si tiene cambios sin fusionar
git checkout nombre_rama Generalmente, Git solo permitirá que nos movamos a otra rama si no tenemos cambios. Si tenemos cambios, para cambiarnos de rama, debemos:
1. Eliminarlos - deshaciendo los cambios
2. Confirmarlos - haciendo un git commit

Guardar cambios y subirlos al repositorio remoto
Una vez que terminamos de realizar los cambios que queremos en nuestra branch, ejecutamos los mismos comandos que vimos hasta ahora: git add, git commit, git status y git log. Pero cuando queramos subir esos cambios, debemos utilizar git push con el nombre de la rama en que estamos posicionados:
• git push origin <branch>
Así también, para traer los cambios de esa rama utilizamos el git pull agregando desde donde queremos traer los cambios:
• git pull origin <branch>

git branch nombre de rama a crear
git checkout nombre de rama para movernos a esa rama

/ = raiz (rutas absoluta si lleva /)
(ruta relativa no lleva /)